$\boxed{\text{F16}}$

* Algoritmer
* Implementation i Matlab
* Stabilitet

Kap 6.3 - 6.4

6.4 Algoritmer

Eulers metod:

$u_i = u_{i-1} + k_i f(t_{i-1}, u_{i-1})$,

f Lipschitz kont i u,

$|u(t_n) - u_n| \le \frac{kM}{2L}\left(e^{L(t_n - t_0)} - 1\right)$, $|u''| \le M$

Vi löser $u'(t) = \cos(u(t))$
med begynnelsevillkor $u(0) = 1$:

```
1: n=10; T=pi; k=T/n;
2: t=linspace(0,T,n+1);
3: u=zeros(1,n+1);
4: u(1)=1;
5: for i=1:n
6:     u(i+1)=u(i)+k*cos(u(i));
7: end
8: plot(t,u)
```

Eulers metod är explicit eftersom u(i+1) ges genom insättning av u(i).

Bakå Euler

$$u_i = u_{i-1} + k_i f(t_i, u_i)$$

Vi löser samma problem

```
n=10; T=pi; k=T/n;
t=linspace(0,T,n+1);
u=zeros(1,n+1);
u(1)=1
for i=1:n
    x=u(i); xold=u(i)+1;
    while abs(x-xold)>1e-8
        xold=x;
        x=u(i)+k*cos x;
    end
    u(i+1)=x;
end
```

Ekvationen $u_i = u_{i-1} + k\cos u_i$ löses alltså med fixpunktsiteration. Metoden är implicit.

Mittpunktsmetoden

```
n=10; T=pi; k=T/n;
t=linspace(0,T,n+1);
u=zeros(1,n+1);
u(1)=1;
for i=1:n
    x=u(i); xold=u(i)+1;
    while abs(x-xold)>1e-8
        xold=x;
        x=u(i)+k*cos((u(i)+x)/2);
    end
    u(i+1)=x;
end
```

Mittpunkt är alltså också implicit.

Axel Målqvist axel@chalmers.se

Den algebraiska ekvationen

$u_i = u_{i-1} + k f(t_i, u_i)$ har inte alltid lösning:

<u>Exempel:</u> $u'(t) = u(t)^2$, $u(0) = 1$

med lösning $u(t) = \frac{1}{1-t}$, obegr. i $t=1$.

Bakåt Euler metoden ger

$u_i = u_{i-1} + k u_i^2$ med lösning

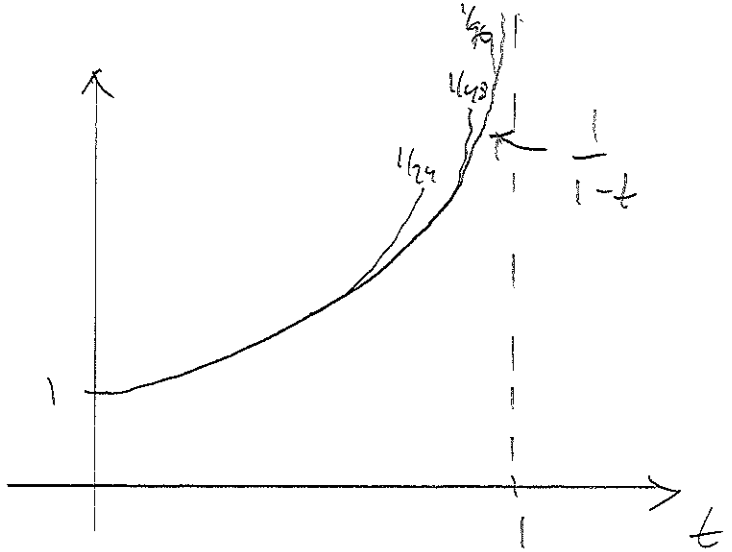
$$u_i = \frac{1}{2k} (\pm) \sqrt{\frac{1}{4k^2} - \frac{u_{i-1}}{k}} = $$

begynnelse villkor $u_0 = 1$

$$= \frac{1}{2k}\left(1 - \sqrt{1 - 4k u_{i-1}}\right).$$

För $i=1$, $u_0 = 1$, ger $\sqrt{1-4k}$ som bara är definierad om $k < \frac{1}{4}$.

När u_{i-1} ökar krävs mindre k



Lösningen existerar olika länge beroende på k.

Datordemo i Matlab.

1) $u'(t) = \cos(u(t))$, $u(0) = 1$,

- Eulers metod
- Bakåt Euler
- Mittpunkt

2) $u'(t) = u(t)^2$, $u(0) = 1$

- Bakåt Euler

t= linspace(0,0.96,100); plot(t,1./(1-t));

3) $u'(t) = -4{,}2\,u(t)$, $u(0) = 1$

- Eulers metod

$k = \frac{T}{80}, \frac{T}{40}, \frac{T}{20}, \frac{T}{10}$

6.3 Stabilitet

Vi ser att Eulers metod blir instabil för snabbt varierande lösningar:

Feluppskattningen

$$|u(t_n)-u_n| \leq \frac{kL}{2M}\left(e^{L(t_n-t_0)}-1\right)$$

innehåller termen $e^{L(t_n-t_0)}$ som blir stor om L och t_n är stora. Lösningen behöver alltså inte alls vara bra.

Från sats 6.1 (beviset) har vi

$$e_i = e_{i-1} + kf(t_{i-1},u(t_{i-1})) - kf(t_{i-1},u_{i-1}) + \frac{k^2}{2}u''(\eta_i)$$

$$= e_{i-1} - k\,4.2\,e_{i-1} + \frac{k^2}{2}u''(\eta_i) \qquad \eta_i \in [t_{i-1},t_i]$$

$$\approx (1-4.2k)e_{i-1}$$

För $k=\frac{5}{10}$ får vi $|1-4.2\frac{5}{10}|=1.1>1$, felet växer alltså med en faktor 1.1 i varje tidsteg.

För $k=\frac{5}{20}$ har vi $|1-4.2\frac{5}{20}|=0{,}05<1$

Om $|1-4{,}2k|>1$ växer den numeriska lösningen exponentiellt, metoden är instabil.

För att avgöra om en metod är instabil för ett givet k används en testekvation.

$$u'(t) = \lambda u(t), \quad \lambda \in \mathbb{C}$$

Eulers metod: med steglängd k ger

$$u_i = u_{i-1} + k\lambda u_{i-1} = (1+\lambda k)u_{i-1}$$

Bakåt Euler:

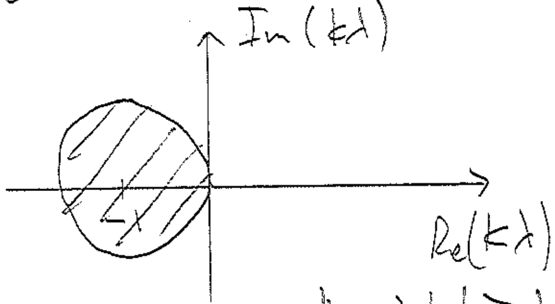
$$u_i = u_{i-1} + k\lambda u_i \Rightarrow u_i = \frac{1}{1-\lambda k} u_{i-1}$$

Mittpunktsmetoden

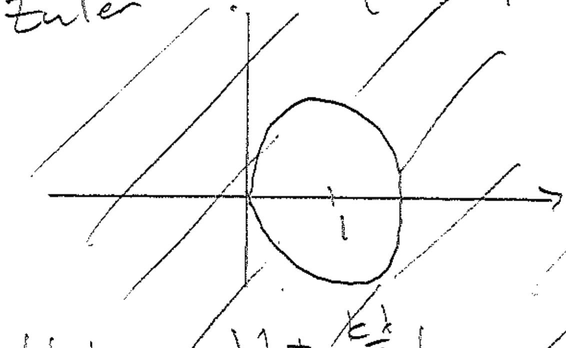
$$u_i = u_{i-1} + \frac{k\lambda}{2}(u_i + u_{i-1}) \Rightarrow u_i = \frac{1+\frac{k\lambda}{2}}{1-\frac{k\lambda}{2}} u_{i-1}$$

En numerisk metod definierad av $u_i = g(\lambda k)u_{i-1}$ är stabil om $|g(\lambda k)| < 1$.

Eulers metod $|1+k\lambda| < 1$



Bakåt Euler: $|1-\lambda k| > 1$



Mittpunkt: $\left|\frac{1+\frac{k\lambda}{2}}{1-\frac{k\lambda}{2}}\right| < 1$